NAT'L INST. OF STAND & TECH
A11106 689917
NIST PUBLICATIONS
REFERENCE

NISTIR 7004

# Master Product Model for the Support of Tighter Integration of Spatial and Functional Design

Steven J. Fenves

Young Choi

Balan Gurumoorthy

Greg Mocko

Ram D. Sriram

QC
100
.U56
NO.7004
2003

NIST
National Institute of Standards and Technology
Technology Administration, U.S. Department of Commerce

NISTIR 7004

# Master Product Model for the Support of Tighter Integration of Spatial and Functional Design

**Steven J. Fenves**
*NIST Research Associate*
*Carnegie Mellon University*
*MEL/MSID*

**Young Choi**
*NIST Research Associate*
*Chung-Ang University, Seoul, Korea*
*MEL/MSID*

**Balan Gurumoorthy**
*NIST Research Associate*
*Indian Institute of Science, Bangalore, India*
*MEL/MSID*

**Greg Mocko**
**Ram D. Sriram**
*NIST/MEL*

May 2003

**U.S. DEPARTMENT OF COMMERCE**
*Donald L. Evans, Secretary*
**TECHNOLOGY ADMINISTRATION**
*Phillip J. Bond, Under Secretary of Commerce for Technology*
**NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY**
*Arden L. Bement, Jr., Director*

## Table of Contents

List of Figures ..... ii
List of Tables ..... iii
1 Introduction ..... 2
2 Background ..... 2
2.1 Current practice ..... 2
2.2 Illustrative usage scenarios ..... 4
2.3 Current research and development ..... 6
3 Conceptual data architecture ..... 11
3.1 The master model ..... 11
3.2 Premises underlying the architecture ..... 13
3.3 Generic nature of the idealization and mapping processes ..... 14
3.4 Additional usage scenarios ..... 16
4 Aspects requiring further research ..... 17
4.1 Characterization of functional models ..... 17
4.2 Potential customization ..... 18
4.3 Enabling opportunistic analysis ..... 18
5 An illustrative example of idealization ..... 19
5.1 The models ..... 20
5.2 Model idealization for functional analysis ..... 23
5.3 Comparison of analysis results ..... 23
6 Summary and conclusions ..... 26
7. Acknowledgements and disclaimer ..... 27
References ..... 28

## List of Figures

Figure 1. Retroactive analysis scenario....5
Figure 2. Integrated spatial and functional design scenario....5
Figure 3. Conceptual data architecture .... 13
Figure 4. Initial idealization operator hierarchy.... 16
Figure 5. Function-driven design scenario .... 16
Figure 6. Multifunctional design-analysis scenario.... 17
Figure 7. Idler arm .... 20
Figure 8. Initial functional model.... 21
Figure 9. Detailed functional model.... 22
Figure 10. Mesh of initial functional model .... 24
Figure 11. Mesh of detailed functional model.... 25

## List of Tables

Table 1. Comparison of key analysis parameters ..... 25

# Master Product Model for the Support of Tighter Integration of Spatial and Functional Design

Steven J. Fenves, Young Choi, Balan Gurumoorthy, Greg Mocko and Ram D. Sriram

## Abstract

In this report, we review current engineering practices and the research literature dealing with the integration of spatial and functional design in the product development process. We then propose a conceptual data architecture that can provide the technical basis for making tighter integration of spatial and functional design, analysis-driven design and, eventually, opportunistic analysis more pervasive. We examine some of the principles that may be used to organize and implement the proposed data architecture, without at this time committing to any one of the principles. We examine some of the outstanding research issues that will have to be addressed before implementation could be undertaken. We illustrate some of the data integration issues with a simple example. Finally, we outline some of the steps toward the acceptance and implementation of the proposed data architecture. Organizational and education/training issues that affect the lack of more intimate integration of spatial and functional design are briefly mentioned as part of the cause of the current lack of intimate interaction, but they are not considered further in the report.

## Keywords

product modeling, information modeling, data modeling, object-oriented modeling, Computer Aided Design, Computer Aided Engineering, Finite Element Analysis, interoperability, simulation

# 1 Introduction

Computer aided design (CAD) and computer aided engineering (CAE) tools have thoroughly penetrated into the design processes of essentially all branches of engineering and manufacturing. These tools are increasingly being integrated within organizations, as well as across organizations on a global basis. These tools are also being increasingly linked to tools supporting other processes within organizations, from marketing through manufacturing to operation, maintenance and eventual disposal. A healthy and highly competitive vendor industry has developed for supplying the CAD and CAE tools for the support of these processes, increasingly in the form of commercial off-the-shelf software (COTS). Software vendors are developing increasingly more comprehensive, integrated tools using vendor-specific internal representations. At the same time, a large number of organizations, NIST among them, is working on data representations and standards that enable interoperation and integration among heterogeneous tools.

This report begins with an investigation of why a particular mode of integration of the spatial and functional aspects of design has become prevalent in a large segment of industry and why other modes of more intimate interaction are not supported as well. It then proceeds to propose a radically different data architecture than currently in use, although one accepted in the database domain over 30 years ago.

# 2 Background

## 2.1 Current practice

In a comprehensive paper on product development, Sellgren states:

> "Geometric models of components and assemblies are carriers of product characteristics. With an unambiguous geometric definition, many aspects of physical behavior are implicitly created at embodiment. ... By adding some extra intents the design model can be transformed to various discretized or lumped behavior models, such as FE-based models, and the performance as well as the reliability of the virtual product can thus be studied and optimized. A solid model is a complete and unambiguous description of the shape and the weight of a body, which is a minimum requirement on a geometric model to be used when creating a model of the physical behavior of that body." (Sellgren, 2000).

Three key issues are embedded in the above statement:

1. The "unambiguous geometric definition" arises only at the end of the embodiment design stage (Pahl & Beitz, 1996) and not earlier;

2. The concept of completeness in "a complete and unambiguous description of the shape" is time-dependent: at any point in the design process the model needs to be complete with respect to the decisions made up to that point, but the complete description of the complete, i.e., finished, design emerges only at the conclusion of a design stage; and

3. Two things have to be done to create the various functional models needed to simulate and evaluate the product's performance (Sellgren calls them behavior models): (a) functional information (Sellgren calls it "some extra intents") has to be added; and (b) the spatial model has to be idealized (Sellgren calls it transformed).

Much of the current practice in the design of mechanical products and assemblies largely disregards issues 1 and 2 above. It is true that in design situations involving speculative and/or

innovative designs, function-to-form transformations and preliminary analyses based on simplified geometric idealizations (e. g., stick models) typically precede the spatial design of the artifact's form or geometry. However, in the vast majority of design situations, typically involving some form of variant or repetitive design, functional modeling and analyses are usually reactive: spatial design, culminating in the complete spatial description down to the most detailed level, precedes all functional analyses. At that late stage in the design process, modifications of the spatial description in response to functional inadequacies revealed by analysis are costly to implement. Prior to that stage, little or no effort is made to capture intermediate spatial descriptions, append to them the "extra intents," perform intermediate functional analyses, and modify the shape as needed to satisfy functional requirements. The current design process described above may result in overdesigned products and/or unnecessary multiple cycles of prototype building, testing and product modification.

This report raises the question: What in current design practice is preventing tighter integration of spatial and functional design throughout the design process, as well as more reliance on analysis-driven design, where the design process is initiated by functional considerations, or on opportunistic analysis, where analyses are initiated as soon as sufficient spatial description becomes available?

Our hypothesis is that four issues underlie the current situation:

- *Organization of design teams:* In engineering organizations with highly specialized personnel, people designated as "designers" are responsible for the artifact's shape, those designated as "analysts" are responsible for ascertaining the artifact's behavior by means of analyses and those called "engineers" are responsible for ensuring that the artifact satisfies all the imposed functional requirements and exhibits adequate behavior. Within this type of organization, the design sequence is often fixed in the order designers – analysts - engineers. Fusion of expert knowledge from designers, engineers and analysts, required for the implementation of tighter integration of spatial and functional design, is frequently lacking.

- *Disparity in tools:* The use of Computer Aided Design (CAD) tools for shape generation is well supported by commercial off-the-shelf products (COTS) and widely accepted in manufacturing enterprises. In contrast, the capture of "engineering" input for analysis-driven design is not well supported: the sketches, cross sections, block diagrams, stick diagrams, etc., used by engineers are generally not accepted as input by current tools either for constructing initial idealized functional models or for using these idealized models to initiate detailed shape descriptions.

- *Disparity in interoperability of tools:* In response to the design-analysis process sequence described above, vendors of most Computer Aided Engineering (CAE) analysis tools provide for the import of shape descriptions from CAD representations. However, the reverse is not true: with today's tools, CAD shape representation cannot be initiated from functional considerations.

- *Domain-specific nature of idealization/mapping between spatial and functional models:* The process of extracting from the CAD spatial model a domain-specific model suitable for a particular functional analysis, which we call *idealization*, involves many decisions specific to that domain. Similarly, the process of updating the spatial model to reflect the results of a domain-specific functional analysis, which we call *mapping*, again involves many decisions, because a single modification of the artifact's shape may affect behaviors in several different functional domains. Thus, the idealization and mapping transformations between the CAD and CAE tools do not have the generality of the individual stand-alone tools, and vendors have been unwilling to implement them.

One further barrier to interoperability in commercial software is the proprietary data representation used by the various software vendors. As interoperability increases across software tools, software vendors worry that customers will purchase products sold by other companies, resulting in revenue loss to the vendors. For this reason, attempts at improving interoperability among heterogeneous tools, such as the incorporation of STEP-reliant representations into current software tools, has received resistance from vendors. Vendors tend to encourage the use of monolithic software suites to address the various aspects of engineering design. However, monolithic software solutions do not solve the full breadth of interoperability problems. It is highly unlikely that a single system that supports all the various engineering design activities will be the best at all such activities.

A technical report, and even an implemented and deployed technical solution, will not overcome the organizational, professional, social and commercial barriers listed above. It is our expectation that a well thought-out and well implemented technical solution can eventually reduce, and in the long run even eliminate, the effect of some of the above barriers. This report is seen as the first step toward that end.

### *2.2* Illustrative usage scenarios

Two typical current usage scenarios are illustrated below.

In the simplest scenario, which we term *retroactive analysis* and which unfortunately is followed much too frequently, the spatial design is completed before any functional analysis is undertaken, as illustrated in Figure 1. A candidate design is advanced through the Preliminary and Detailed Design stages based upon designer experience and knowledge. The design is validated by means of a functional analysis only at the completion of the Detailed Design stage. To accomplish this, the analyst idealizes the spatial model by eliminating information not needed for analysis, either due to limitations of the analysis tools or because the mass of detail may hide important aspects of behavior. If the analysis results are satisfactory or nearly satisfactory, the designer completes the final design as planned. If the analysis results are unsatisfactory, the designer must return to the Detailed Design stage - or even the Preliminary Design stage - to resolve the deficiencies. Another iteration of spatial redesign followed by functional analysis then follows. This retroactive analysis approach may result in overdesign - to reduce the expense or delay of design iterations caused by inadequate behavior revealed by analysis - and/or in multiple cycles of prototype building, testing and modification.

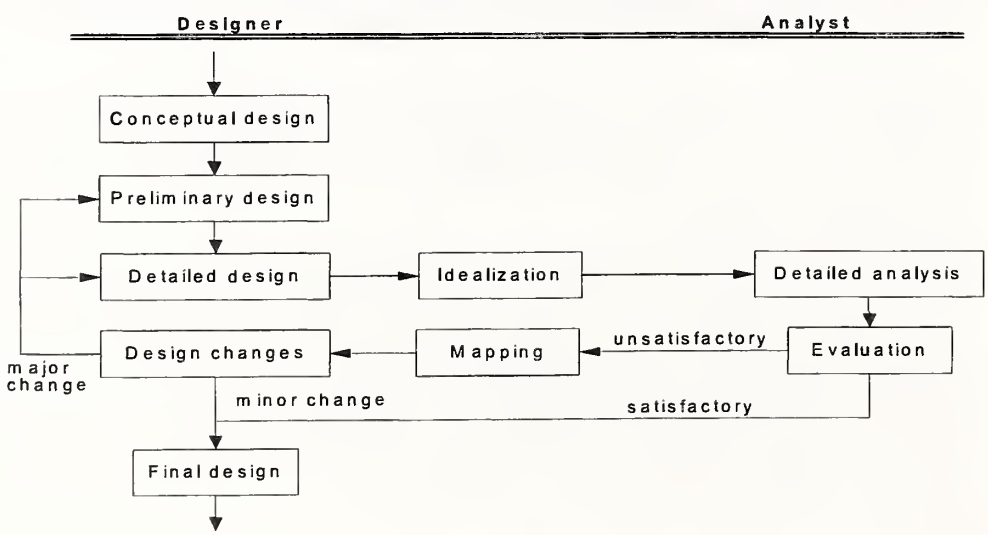

**Figure 1. Retroactive analysis scenario**

A more proactive, *integrated spatial and functional design* scenario involves analyses at earlier stages so that analysis results may guide subsequent design decisions, as illustrated in Figure 2. In this scenario, the design is analyzed repeatedly throughout the design process. After completion of the Preliminary Design stage, the spatial model as it exists at that stage is analyzed to ascertain its functionality. If the behavior is satisfactory, spatial design proceeds to the Detailed Design stage with confidence. If not, the preliminary spatial design is revised and the functional analysis reiterated. Due to the absence of detail at the Preliminary Design stage, idealization of the spatial model is greatly simplified and design modification/analysis iterations can be performed quickly. In the Detailed Design stage, the necessary detail is added to the spatial model, significantly expanding both the complexity and size of the product model. At the completion of the Detailed Design stage, the design is analyzed again. Much of the added design detail must be suppressed and the functional model idealized to make it suitable for analysis. If the analysis predicts satisfactory or near satisfactory behavior, the design may be completed. If not, the Detailed Design stage must be repeated; it is highly unlikely that the Preliminary Design stage will have to be revisited to correct the deficiencies. The chance of major spatial redesigns indicated by functional analyses of detailed designs can thus be significantly reduced.

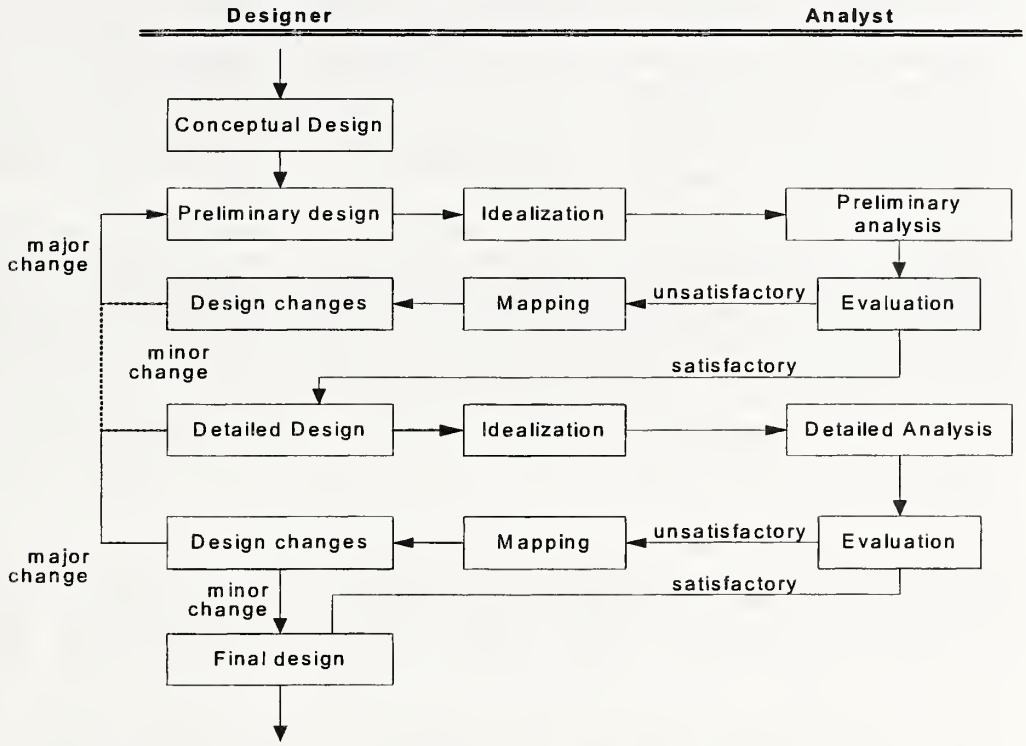


**Figure 2. Integrated spatial and functional design scenario**

Additional usage scenarios made possible by the proposed data architecture will be presented in Section 3.3.

## 2.3 Current research and development

The current state of research and practice in integration of spatial and functional design is described in four categories: (1) modular systems; (2) Computer Aided Design and Finite Element Analysis (CAD-FEA) integration; (3) product modeling and data exchange efforts; and (4) multi-aspect information structures.

### 2.3.1 Modular systems

Modular systems based on object-oriented modeling principles have the benefit of reusability and high modularity; they also provide an intuitive way to model real-world systems. Three representative research projects are discussed: (1) The Composable Simulation project; (2) The MOSAIC project; and (3) The Multi-view Representation Architecture.

The Composable Simulation project at the Institute for Complex Engineered Systems at Carnegie Mellon University is intended to automatically generate system level simulations from individual components (Diaz-Calderon, 2000; Paredis, 2001; Sinha, 2000; Sinha, 2001). The project uses the concepts of reconfigurable models and component libraries. A *reconfigurable model* is a system representation based on interface and implementation, where interface describes the model's interaction through ports and implementation describes its internal behavior. Simulation models are composed in an object-oriented, hierarchical fashion from model fragments; multiple models may be associated with a single system component - these models can be easily reconfigured (through composition and instantiation) to suit a particular simulation experiment. Ports represent the localized points of interaction on the boundary of a model and allow for energy flow in and out of the model to interact with other models. The *component library* is a set of reconfigurable models for use by the designer/analyst.

The MOSAIC project at the Royal Institute of Technology in Sweden, aims to improve the integration of modeling and simulation during the design process. The project has developed a general product model and a prototype system to support the design and simulation of complex products (Andersson and Sellgren, 1998). The MOSAIC system treats the product development process and the engineering data created by it as technical systems. A product is divided into a number of subsystems, which are to be analyzed. Each of the systems can be characterized by what is within its boundaries and how it interacts with other systems. Interfaces between systems are described by mating features and interface features. Mating features represent *what* is connected between two systems and the interface features show *how* the two systems are connected.

The multi-view representation architecture (MRA), under development at the Engineering Information System Laboratory (EISLab) of the Georgia Institute of Technology, is based on information-intensive mapping between spatial and functional models and is aimed at providing the following links: (1) automation of routine analyses; (2) representation of associativity between the spatial and functional models and of the relationships among the models; and (3) provision of appropriate functional models throughout the life cycle of the product (Peak et. al., 1993a; 1993b; 1998). The system consists of four sets of building blocks: (1) Solution Method Models (SMM) are low-level, solution-specific methods that combine inputs, output, and control for a single type of analysis for diverse solution methods and for various vendor-specific tools; (2) Analysis Building Blocks (ABB) represent engineering concepts that include engineering semantics; they are constructed utilizing constraint graphs and object-oriented techniques; and they can be assembled to form analysis systems representing a particular model; (3) Product Models (PM) represent all data associated with a product over its life cycle and can form reusable functional models based on idealizations and simplifications; and (4) Product Model-Based Analysis Models (PBAM) contain the linkages between a PM and the product-independent ABBs to solve specific analysis problems.

The first two projects provide valuable insights into interaction and interoperability issues; however, at the present, they do not appear to accommodate commercial off-the-shelf software (COTS) as modules or components, and are therefore not suitable for the large number of organizations that depend primarily, if not exclusively, on COTS tools. The third project is specifically conceived with COTS analyzers in mind; however, it does not provide explicit spatial and functional models of the artifact.

### 2.3.2 Computer Aided Design and Finite Element Analysis integration

Research on CAD-CAE integration has largely concentrated on a single CAE analysis tool, namely, Finite Element Analysis (FEA). CAD-CAE integration research can be categorized into two areas: (1) the *microscopic* view dealing with issues such as automatic mesh generation, model simplification, loading and boundary condition idealization, etc., required for creating finite element models; and (2) the *macroscopic* view concerned with overall product data structuring and with sharing and reuse of product data.

An illustrative microscopic approach is presented by Armstrong, et al. (Armstrong et al., 1996), expanding on the concepts described in (Armstrong, 1994). The approach defines operations that allow analysts to suppress detail and reduce the dimensionality of a part so as to remove geometric features that cause disturbances in the stress field, resulting in simpler meshes and faster FE analyses. While these operations are important to assist in the creation of functional models based on design models, they require manual intervention by the analyst and do not depend on any associativity between the models.

The macroscopic view of CAD/FEA integration focuses on the formal description of products on standards to support integration of product information.

Arabshahi, et al. (Arabshahi et al., 1991) present a vision of CAD and FEA-based integration of spatial and functional design. As an extension, Arabshahi, et al (Arabshahi et al., 1993) present an automated CAD-FEA transformation method. The overall aim of the work is to enable analysis to respond to changes in the design and also to allow seamless integration between spatial and functional design and analysis. Belaziz, et al. (Belaziz et. al., 2000) develop a feature-based tool to aid in the integration of analysis during the design process based on a morphological analysis of solid models.

Sellgren and Drogou (Sellgren and Drogou, 1998) develop an object-oriented approach to FEA modeling separating models, submodels, interfaces and orientations. Systems are described by recursive subsystems and related through interfaces. A system model, an idealized representation of a system at a level of complexity and detail suited to the analysis at hand, is an aggregated model of subsystems connected by interfaces. The interfaces are aggregates of mating faces. Behavior features represent the form features at a particular level of detail and describe the physical properties of the form feature and how the feature is related to other features. Form features can be represented by a number of different behavior features for different fidelity models.

Shapiro (Shapiro, 2002) generalizes the issue by using the same normalized functions both for describing the shape and the interpolation functions used in FEA. Such functions may be constructed automatically for most types of spatial models using the theory of R-functions. The interpolated boundary conditions are combined with a set of basis functions, typically on a uniform rectangular grid, into a solution structure for the problem at hand that is assembled and solved at run time.

Gordon provides a concise assessment of the integration problem: "Today's bottleneck in CAD-CAE integration ... lies with efficient creation of appropriate simulation-specific geometry (in our terminology, discipline-specific)" (Gordon, 2001). Gordon proceeds:

The key to understanding CAD-CAE Integration is related to the scale, scope and purpose of the required engineering analysis, e.g., Finite Element Analysis (FEA). The closer the scale, scope and purpose of an engineering analysis is to the type and detail of the existing CAD product model geometry, the greater the likelihood that a closely-coupled, automated, or even seamless integrated CAD-CAE process can be implemented. Whether CAD-CAE applications can be closely-integrated and automated depends upon:

- The scale, scope, and purpose of the CAE analysis.
- The nature and type (order, or "gender") of the captured CAD geometry.
- The amount of detail required for the CAE application" (Gordon, 1997).

Gordon proceeds to provide the following categories of CAD-CAE interactions:

- Category I - The CAD geometry and the functional analysis-specific geometry are the same. This is the truly "seamless" case; there is no change in detail, no de-featuring, and no geometry order changing required. Analysts and designers use the same (or duplicate copies of the same) geometry.
- Category II - Existing (available) CAD geometry has the wrong content; it is too detailed and/or of the wrong type to support the scale, scope, and purpose of the required or most appropriate type of analysis. Changes are required to add features or remove unnecessary detail from, and/or modify the order of, the CAD geometry to create a domain-specific geometry amenable to analysis. Automated and semi-automated procedures are required.
- Category III - Engineering analyses are performed first to define and refine a design concept using idealized geometry prior to establishment of the enterprise (CAD) product model. Domain-specific geometries for functional models will require modification and the addition of details and features to support full spatial design and manufacturing. Automated and semi-automated procedures are desirable (Gordon, 1997).

The papers in this category provide many concepts useful for building a repertoire of idealization techniques. Gordon's presentation provides a framework for CAD-CAE interactions that are valuable beyond the specifics of FEA-based analyses.

### 2.3.3 Product modeling and data exchange efforts

There is a strong trend in current research and practice toward the standardization of product information. A large effort has been directed to the development of ISO 10303 standards for data exchange. ISO 10303, commonly known as STEP, aims to eliminate many of the problems associated with integration by providing a method for the platform-independent sharing and exchange of product data information (ISO, 1994a).

STEP includes AP 209, which deals with Composite and Metallic Structural Analysis and Related Design (ISO,2001). AP 209 covers the product definitions of the analysis and design disciplines. The shape representation in AP 209 is interoperable with that in AP 203 (ISO, 1994b), which is currently implemented in many commercial CAD applications. Although the focus of the integration of spatial and functional design cannot be limited to form alone, form does play a major role in each domain. AP 209 provides an important mechanism for sharing geometric information between functional models and design models and aims to provide a standards-based solution for the integration of spatial and functional design (Hunten, 1997). As stated below, one of the major roadblocks in integration of spatial and functional design is the generation of the appropriate geometry for functional analysis. Hunten (Hunten, 2001) discusses the geometric

transformations available within AP 209 and emphasizes the ability to link the idealized analysis shape to the actual design shape by the "based-on" linkage.

The Engineering Analysis Core Model (EACM) is part of the STEP standard suite (Leal, 1999). The goal of EACM is to increase the scope of STEP by capturing all engineering information to support business practices. EACM deals with three key aspects: (1) the management of engineering analysis and design information; (2) the linking of the information to activities, decisions, and analyses; and (3) the storage and retrieval of information. These three characteristics of EACM bridge the gap between CAD, Product Data Management (PDM) and analyses. Modules in EACM will provide data management information, the definition of properties, audit trails for product information, and mathematical techniques (Leal, 1999).

STEP AP 209 provides a valuable linkage or associativity between spatial and functional models, albeit one-directional (the analysis shapes may be "based-on" design shapes, but the inverse linkage is not maintained). The EACM illustrates some of the additional "intents," to use Sellgren's term, that may be associated or linked to the engineering model.

The studies reported provide useful information on the variety of transformations between spatial and functional models and on their representation and exchange. It is to be hoped that accumulated knowledge from the mature field of Finite Element Analysis can be emerging functional analysis fields such as computational fluid mechanics.

### 2.3.4 Multi-aspect information structures

This section provides an overview of research on multi-aspect, hierarchical information structures.

Three models are described: (1) the Product Master Model; (2) the Pluggable Metamodel Mechanism; and (3) the Multiple View Intermediate Modeler

While unifying concepts and implementations vary widely, the three models share the property that the information models representing the various discipline-specific aspects of the artifact are linked, physically or virtually, to a master model that contains all the information about the artifact.

Hoffman and Joan-Arinyo (Hoffman and Joan-Arinyo, 1998) present the Product Master Model, an architecture that keeps consistent associations between CAD and downstream applications. The authors raise crucial questions on the issue of integration:

> "...the data in the master model originate from different domain-specific programs; how can this information be kept consistent and how is it maintained under design changes? In our view, the CAD system is one of the clients of the master model, with the primary charge of creating and maintaining the net shape information. ...
>
> How can we establish and maintain a persistent association between the geometry data contributed by the CAD system and data originating from other application programs?" (Hoffman and Joan-Arinyo, 1998)

The master model is an object-oriented repository that has mechanisms for maintaining the integrity and consistency of the information structures. The master model serves several clients, one of which is the CAD application, responsible for creating the initial net shape and also for modifying the net shape. For each additional client, there is a corresponding view of the product. Each client application can post product information it processes to the master model, as well as keep a private repository relevant to itself. When a change is made to the model, a protocol is followed to ensure that the most up-to-date product data is available to all clients. The change

information is posted, and it is up to the clients to re-associate with the new information. The overall architecture of the master model is an object server in charge of coordinating the information to all the clients.

A net shape associativity mechanism creates and maintains the associativity between elements in the net shape of the product. Once the net shape is deposited in the master model, each client application is allowed to associate information with it. For each net shape, the master model creates an inventory of the applications that have made an association to it. If the net shape is changed in the CAD system, the master model calls on each of the applications that are associated with the element.

Hoffman and Joan-Arinyo summarize the difficulties with the product master models. The biggest problem is the ability to establish association between net shapes and to keep the information consistent in a distributed network. Additionally, as previously stated, problems arise from the proprietary data and methods of engineering applications. However, the authors believe that the change protocol provides a realistic mechanism for associating downstream applications to the CAD model. In principle the authors' approach is valid, but there are a number of potential problems with the change protocol mechanism. Each time the model is changed, a series of "exports" and "imports" must be completed in both the CAD and the CAE parts. Although the changes can be propagated in the geometric sense, there is still a discontinuity in the shape and the functional models, unless the system is restricted to deal only with Gordon's Category I described above.

Yoshioka and Tomiyama (Yoshioka and Tomiyama, 1997) present a mechanism for integrating various aspect models, such as geometric, kinematic, and finite element models, for knowledge intensive engineering. Their Knowledge Intensive Engineering Framework (KIEF) is constructed using multiple aspect models related through a metamodel mechanism. The metamodel represents the relationships between the concepts in the aspect models. The framework aims to integrate and maintain the consistency of the various models in all of the product phases. The KIEF framework integrates commercially available software tools through the Pluggable Metamodel mechanism.

An aspect model is a model of an artifact from a particular point of view. An aspect model is built by first constructing relationships between it and other existing aspect models. Model simplification and abstraction are common tasks during this step. Next, data is transferred from the existing models to complete the new aspect model. The metamodel describes how information is exchanged among the aspect models. However, it is not always easy to extract all the necessary parameters to complete an aspect model. For this reason, the ability to plug in existing modelers is presented. Finally, mechanisms for representing data and selecting appropriate modelers are presented. The technology presented by Yoshioka, et al. contribute to the master model paradigm of product design significantly. However, it is not clear at the present how the various modelers actually share product information to support the various aspect models.

De Martino, et al. (De Martino et al., 1998) present an approach to CAD-CAE integration based on design-by-features and feature recognition. Feature-based modeling allows for the representation of semantic information about the product and for more direct communication between design and engineering processes. However, the sharing of semantic information across engineering applications and domains is not currently supported.

To achieve the integration between the processes, a common model must exist, providing different views for different analysis domains. The Intermediate Model (IM) is shared between applications and provides them with context-specific feature-based views. Initially, the designer creates an object using design features from a library. The design is evaluated and stored in the IM to maintain the semantics of the features. Feature recognition "templates" are available for

each of the analysis domains. An integrated feature-based system links design and engineering activities.

The kernel activities consist of design-by-features, solid modeling, feature recognition, and feature matching. Design by features is based on the parametric instantiation of features retrieved from a library. In solid modeling, geometric evaluation and any needed change of the geometry is performed. During the shape feature recognition process, the algorithm iterates to recognize the shapes based on context independent information and returns a neutral file. Finally, in the application feature matching process, the generic shape features extracted from the shape recognition process are interpreted into context-dependent features.

The intermediate model provides the main source of information for the various processes. The processes maintain the global state and information content of the intermediate model. An important problem with representing features from different points of the view is that of feature interaction and shape sharing. Additionally, the features may have different representations in different domain views. The intermediate model supports the existence of different representations and non-homogenous data.

The three projects reviewed address in a direct way the integration of spatial and functional design issues at the core of this report, and provide three alternatives for structuring and implementing the needed relationships between the spatial and functional models.

A more extensive discussion of the works cited, as well as survey of additional integration of spatial and functional design issues may be found in (Mocko & Fenves, 2003).

# 3 Conceptual data architecture

## 3.1 The master model

It is clear that meaningful integration of spatial and functional design requires a master model of the product being designed and an arbitrary number of specialized domain-specific functional models interfaced with or served by the master model. The master model acts as the persistent representation of the evolving product, maintaining consistency and integrity of product data across the different functional domains. Such a multi-aspect architecture is embodied in the Product Master Model of Hoffman and Joan-Arinyo (Hoffman and Joan-Arinyo, 1998), the Intermediate Model of DeMartino et al. (DeMartino et. al., 1998) and, in a virtual form, in the metamodel of Yoshioka and Tomiyama (Yoshioka and Tomiyama, 1997).

The conceptual multi-aspect architecture has a long-standing antecedent in the well-established "three-schema architecture" of Database Management Systems (DBMS) (CODASYL 1971, ANSI/SPARC 1975): the master model serves as the global schema (alternate terms: logical, conceptual schema) for all information about a product being designed, while the models for the various functional analysis domains constitute views (alternate terms: user views, external schemas); the physical storage schema completing the three-schema architecture is abstracted out and will not be considered further.

The two key issues concerning the multi-aspect architecture are:

1. What should be the organizing principle for structuring the information; and
2. What principles should govern the implementation.

These two issues are discussed in the following sections.

### 3.1.1 Organizing principle

Hoffman and Joan-Arinyo propose a net shape associativity mechanism as the means of linking non-geometric functional information to shape elements. This is a plausible organizing principle, provided that it accommodates aggregate or composite shapes as well as the primitive shape elements illustrated in (Hoffman and Joan-Arinyo, 1998). The strongest objection to using shapes as an organizing principle is that the architecture should be universally applicable, even in the absence of shape information. The proposal by De Martino, et al., to use features as the organizing principle can be faulted on two similar issues: functional information may have to be associated with semantic entities more abstract than features, and information needs to be organized even in the absence of assigned features. Yoshioka and Tomiyama do not describe any specific organizing principle for representing relationships between the concepts in the aspect models in the metamodel.

We recommend that the organizing principle, the topmost abstraction level, of the master model be the Core Product Model presented in NISTIR 6736 (Fenves, 2000). In this model, a product is represented by a hierarchy of entities of the class **Artifact,** which is an aggregation of **Function, Form** and **Behavior: Function** represents what the artifact is supposed to do; **Form** represents the proposed design solution for the design problem specified by the function, and **Behavior** represents how the artifact implements its function. **Form** itself is the aggregation of **Geometry,** the spatial description of the artifact, and **Material,** the internal composition of the artifact. Lower-level Artifact entities may be labeled as **Features.** Limited empirical studies at NIST have confirmed that the Core Product Model can serve as the organizing mechanism for a range of design-related information structures. Should it be desired to extend the master model to encompass management-related data as well, as in the EACM proposal, the Core Model could be extended to encompass Activity and Actor classes as well. An even better alternative in this case would be to interface the Core Model with the process Specification Language (PSL), which already supports such classes (Schlenoff et al., 2000).

The two key processes linking the master model to the various domain-specific functional models are idealization and mapping. *Idealization* abstracts from the master model the specific information needed to create a particular domain-specific view. *Mapping* is the opposite operation: it takes the domain-specific information from a particular view and merges it into the master model. These operations are further illustrated with a user scenario in Section 3.3.

The conceptual data architecture is shown in Figure 3, together with the two key processes of idealization and mapping. Unlike present-day CAD/CAE integration schemes, the spatial (CAD) model supporting the design of the Form has no privileged status: it is just one of the specialized "functional" models, operating on its own idealization of the master model by abstracting out Material, Function and Behavior. Unlike traditional DBMS, the master model need not be the union of all views (external schemas): domain-specific properties of functional idealizations that are not relevant to other views may reside only in their respective local models. Finally, again in contrast to traditional DBMS, the maintenance of complete consistency between the master model and all views may not be practical, or even possible. For illustration, assume that on the basis of some functional analysis a mapping involving a modification of the form is warranted. To be fully consistent, the modified form should be re-analyzed to ascertain that the artifact's behavior is now consistent with respect to the functional requirements. Such reanalyses are seldom, if ever, done. Consistency is thus only maintained to the extent that the relationship of the form modification to the analysis that triggered it is recorded in the design record or notebook.

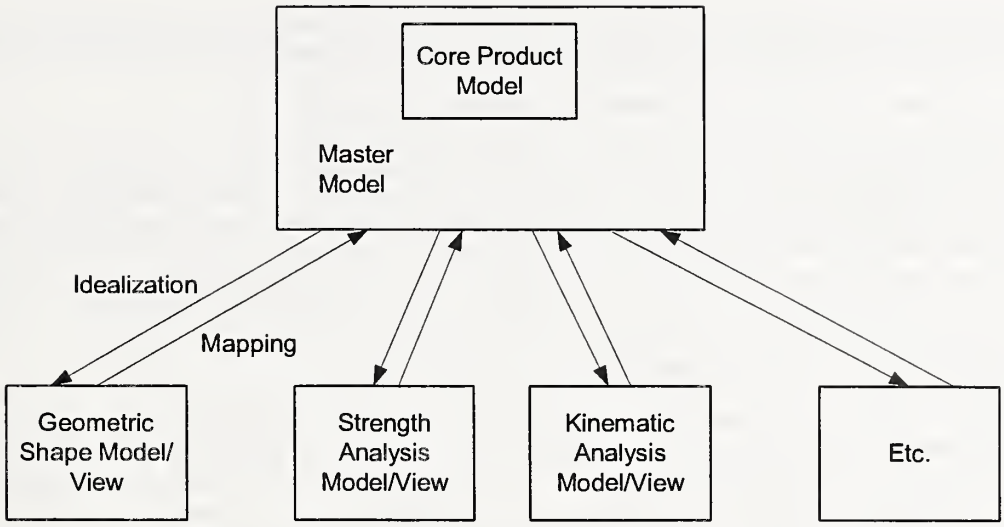


**Figure 3. Conceptual data architecture**

### 3.1.2 Implementation principles

At this stage of the research, we are not in a position to discuss the implementation of the actual data model: further research, limited development and pilot usage will have to determine key aspects of the implementation. One aspect of the implementation is clear: the system will have to be distributed, rather than centralized. However, it is not clear yet what mechanisms will be used by the idealization and mapping processes.

Even at this early stage, the metamodel mechanism of Yoshioka and Tomiyama (Yoshioka and Tomiyama, 1997) appears to be very promising as an implementation principle. The idealization and mapping operations or transformations may be readily viewed as relationships between the concepts in the master model and the concepts in the various aspect models. The "based-on" relationship between shapes in AP 209 is an excellent illustration of such a relationship.

At the present, it is not clear how far the present research should go in the direction of implementation. Possibly, this research should only be concerned with data architecture and interface specifications that may be submitted to the Object Management Group (OMG) (OMG, 2003), the International Standards Organization (ISO) (ISO, 1994a) or other information sharing and standardization organizations.

## 3.2 Premises underlying the architecture

The conceptual data architecture presented above is based on the following two premises.

*Generalizability.* The relationship between the master model and the domain-specific functional models is such that it is possible to idealize functional models for all needed functional domains from the information in the master model and to map back to the master model the updated functional models. At the present, it appears that only a control block diagram model of a mechanical system cannot be easily extracted from master model.

*Design sequence independence.* The design process can be initiated from any one of the functional domains and the master model can be initialized from the information mapped to it from any of the functional models. In this respect, spatial design (traditional CAD) is just another "functional" design domain; thus, traditional CAD is an instance of initiating design from a

"functional" domain. The CAD model generated is just another view that can be mapped to the master model. Some illustrative usage scenarios are presented below.

### 3.3 Generic nature of the idealization and mapping processes

A third underlying premise is that the idealization and mapping processes, well-developed in the CAD-FEA integration area, are generalizable across all the functional domains of interest. As stated previously, idealization refers to the process of obtaining a functional model specific to a particular domain from the master model, and mapping is the inverse process of updating the master model based on changes in the functional model of the domain. Both idealization and mapping can be thought of as conversion processes -- form, function and other aspects of the master model are converted to the appropriate representation required by the functional domain (idealization) and vice-versa (mapping).

Mapping, however, includes a further task of verifying the consistency of the functional information with the master model. Mapping is used to reconcile changes in the functional/domain view with the master model at each step. The results of analyses may be mapped to the behavior in the master model to verify its consistency with the intended function. However, any change in form resulting from analysis has to be checked for consistency with the form in other domains through the form in the master model. Once consistency is verified, associativity ensures automatic updates of form models in every functional domain.

Therefore, in principle, there will be many pairs of transformations (arrows in Figure 3) linking each functional domain model with the master model with each pair of arrows connecting the Function, Form and Behavior, respectively, in the master model with that in the functional domain model. This implies that each view shown in Figure 3 defines a *domain specific functional product model* for a functional domain. For instance, the functional model for the strength analysis domain consists of: an analyzable model (Geometry and Material); meshing information, boundary conditions, and load cases; and analysis results. In this case, the analyzable model and mesh would link to the Form in the master model; the boundary conditions to both the Form and Function; and the results to Form and Behavior.

Having established that idealization and mapping are transformations between different entities in the master and functional models, the remainder discussion deals primarily with mapping and idealization of the form description.

The form model is referred to as geometric or iconic based on the nature of the representation. This classification is based on the basic entities that constitute the form model. The boundary model, features and the analysis mesh are examples of geometric representations where basic geometric entities are used to represent form. Line diagrams representing kinematic chains or block diagrams are examples of iconic representations.

The transformations required to perform mapping and idealization can be categorized into three types, based on Shah's classification of feature transformations: projective, adjoint and conjugate (Shah and Mäntylä, 1995). Projective and conjugate transformations are useful between two geometric form models, while the adjoint transformation handles idealization or mapping between a geometric form model and an iconic form model.

*Projective transformation* involves a change in the dimensionality of the representative entities (3D to 2D, 2D to 1D). For example the mid-surface of a 3D object is used for simulation of injection molding. Medial Axis Transform (MAT) is another idealization that falls under this category. This type of transformation is ideally suited when the domain is not likely to modify the form (no mapping of form required). This is because inverse projective transformation is usually not well defined. While transformations through the use of MAT are reversible (given the MAT it is possible to reconstruct the object) small perturbation to the original geometry result in large changes in the MAT.

*Conjugate transformations* are those that result in form models where the underlying entities constituting the representation are the same, only their arrangement is different, as for example in transforming between design features and machining features.

*Adjoint transformations* handle associativity between a geometric form model and an iconic form model. Typically, the elements are linked through pointers or a look-up table. For example, the links of a mechanism represented by lines in the mechanism functional model can be linked to canonical Boundary Representation (Breps) or Constructive Solid Geometry (CSG) primitives that represent the form in the master model. Changes in the functional domain are assumed to affect only the parameters associated with the icons (lines) and these would translate to parameters used to define the full spatial model.

Feature suppression, or detail removal, which is of particular importance in analysis, does not fit neatly into any of the three categories. The features or detail to be suppressed are however identifiable by a conjugate transformation. A conjugate transformation could yield the features that to be suppressed and a suppression history could be maintained for playback during mapping.

In summary, the premise is that the idealization process is representable as a tree (possibly, a lattice) of idealization operators, similar to a CAD construction history tree: starting with the forms constituting the master model, each idealization operator transforms its input form(s) into its output form(s); the terminal set of idealized forms constitutes the idealized model suitable for analysis. If, in response to analysis results, the shape of some idealized form(s) is (are) modified, the modification can be mapped to the master form by the reverse traversal of tree. All of the concepts and tools of constructive geometry can be brought to bear to correct/heal/adjust any resulting spatial conflicts. The mapping process can be substantially simplified if the process convention is accepted that only the spatial model provides means for changing the Form, a convention adopted by Hoffman and Joan-Arinyo (Hoffman and Joan-Arinyo, 1998). It is understood that some idealizations are not reversible; for such cases, the DBMS analogy to conceptual schema vs. domain views provides links to a rich DBMS literature on what can/cannot be updated *via* views. An initial hierarchy of idealization operators is shown in Figure 4. The "geometric" operators of Merge and Split are included because they are frequently employed in the idealization process. This premise also defines the form representation needed for an analysis driven design scenario: a set of forms (linked as needed by "geometric" operators) that can be mapped into the master model. At this point we are not ready to address the issue of conflict resolution when two independent analyses initiate the design process.

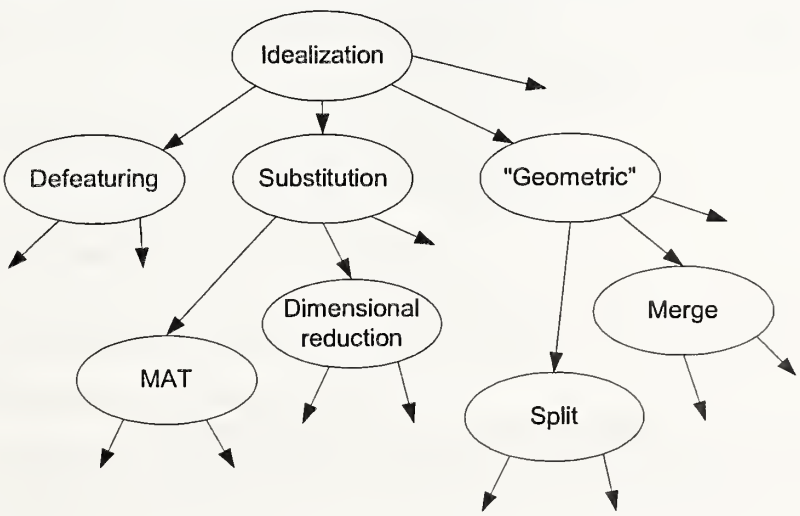

**Figure 4. Initial idealization operator hierarchy**

### 3.4 Additional usage scenarios

In the *function-driven design* scenario illustrated in Figure 5, the design process begins with the specification of the artifact's required Function. From that point, function-to-form transformations and preliminary functional analyses based on simplified spatial representations in one functional domain may be used to populate the initial Form of the artifact, thus enabling both spatial design and further functional analyses to proceed. This Form becomes the starting point for the product designer, who proceeds through the preliminary and detailed design efforts in much that same manner as in the integrated spatial and functional design mode described in Section 2.3. As the design process progresses, *idealization* is used to extract from the Form (Geometry + Material) and Function aspects of the master model the information needed to create models for any kind of domain-specific functional analysis. At the conclusion of each analysis, *mapping* is used to make the master model consistent with the functional model (as a minimum, some analysis results will be mapped to Behavior for comparison with Function; more importantly, modifications of the Form resulting from the functional analysis may also be involved).

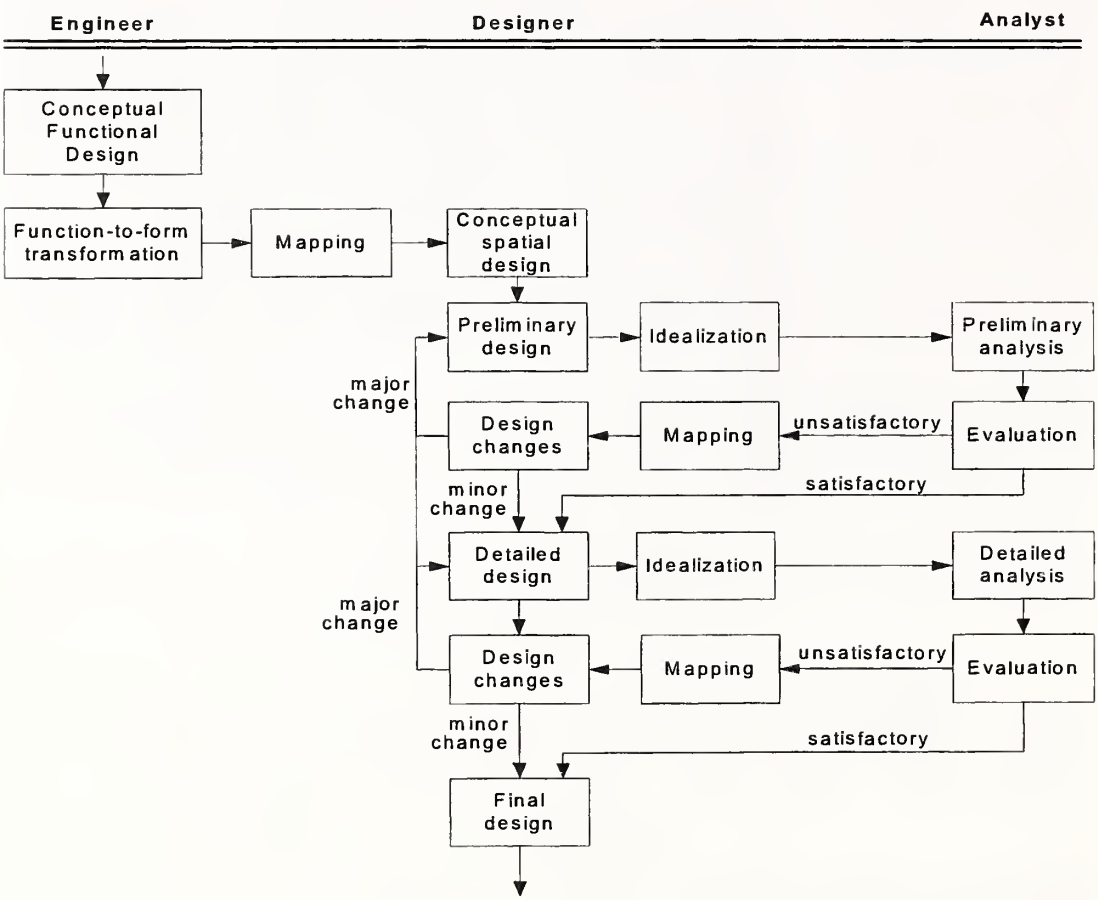


**Figure 5. Function-driven design scenario**

Finally, *multifunctional design-analysis* (involving structural, thermal, dynamic, etc., functional models) may be introduced, in parallel or sequentially, resulting in the scenario shown in Figure 6. Each box labeled "analysis" represents a cycle of idealization, functional analysis, evaluation, shape modification and mapping in one functional domain. As the figure illustrates, multiple domain-specific functional analyses may be performed either within a design iteration cycle or between cycles.

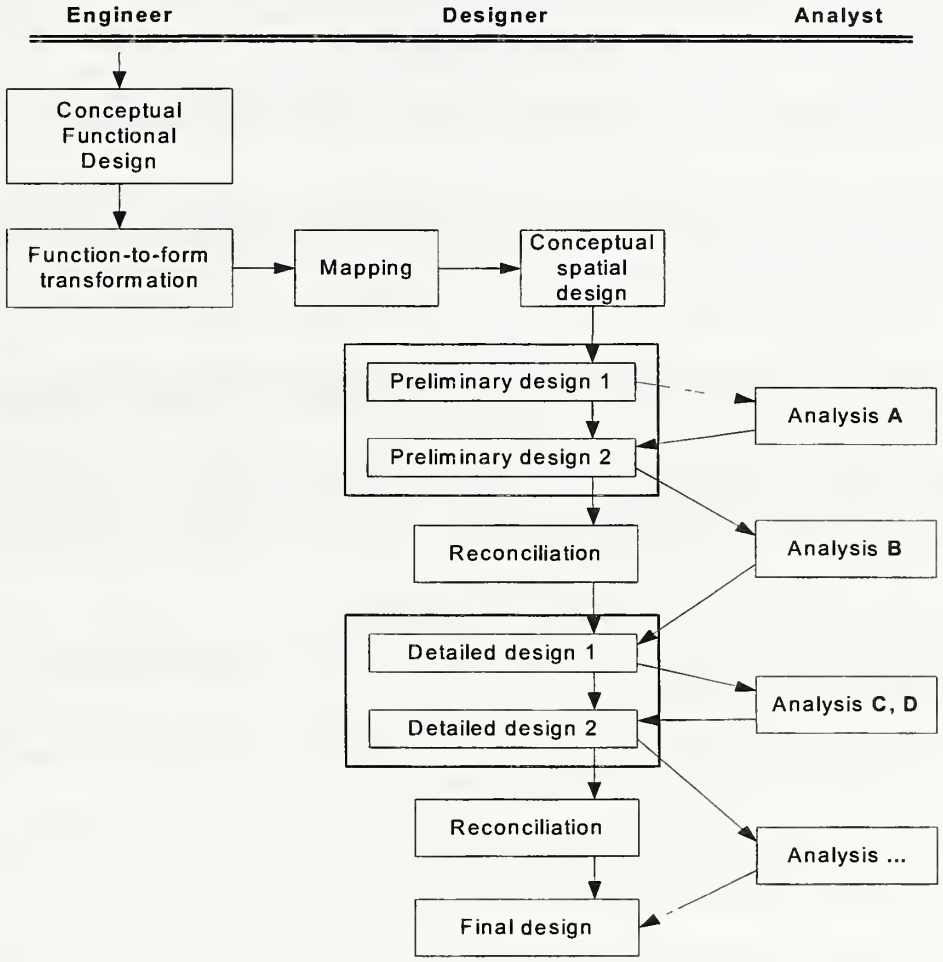


**Figure 6. Multifunctional design-analysis scenario**

## 4 Aspects requiring further research

In this section, we discuss a number of issues that require further research before the conceptual architecture can be expanded into a formal information model.

### 4.1 Characterization of functional models

The various functional models idealized from the master model will have to be characterized according to multiple dimensions. As described earlier, Hunten uses *Scale*, *Scope* and *Purpose*

as three classifiers of the functional models (Hunten, 1997). An initial set of bases for characterization, and an illustrative set of terms in each basis, is as follows:

- *Domain* – e.g., structural, thermal, kinematic, dynamic
- *Intent* – e. g., synthesis aid, behavior evaluation/verification, optimization, repository
- *Fidelity* - level of detail and/or accuracy of results expected
- *Granularity* – e. g., system, component, subassembly
- *Process stage* – e.g., conceptual design, preliminary design, detailed design, manufacturing.

Our *Intent* and Hunten's *Purpose* are probably synonymous; our *Fidelity* and *Granularity* are probably subdivisions of Hunten's *Scale*; and it is not clear from the document what Hunten's *Scope* is intended to cover.

### 4.2 Potential customization

The proposed architecture is of little use if it is not expandable and customizable for all the domains of functional models and their corresponding idealization and mapping relationships or operators.

A possible form of customization is a set of user-defined tables that define the relationship or drive the operators. The concept is illustrated here for the extraction of an idealized functional model from a more detailed master model.

- Rows of the table correspond to the product's features
- Columns of the table correspond to the terms in the various characterization bases of the functional models sketched above
- Each entry in the table denotes the action to be taken for the combination of feature and characterization term, and specifies one of the possible actions:
  - Definitely suppress feature in functional model
  - Consult analyst whether to suppress feature
  - Consult analyst whether to include feature
  - Definitely include feature
  - Warn analyst: Feature should not be present in the master model.

Further experimentation is needed to determine how best to combine the terms of the various characterization bases and whether the form sketched is applicable to all the functional domains of interest and to mapping as well as to idealization.

### 4.3 Enabling opportunistic analysis

Opportunistic analysis means that analysis is initiated as soon as the opportunity for doing so arises, that is, as soon as the information collected in the master model constitutes an idealizable functional model in some domain. As a simple illustration, a structural analysis may be initiated as

soon as the following are defined: the position and magnitude of the applied loads; the position and nature of the supports (boundary conditions); and a sufficient number of structural elements so as to form a continuous "load path" from the applied loads to the supports. Of course, the situation is generally more complex. In particular, in the design of mechanical assemblies, as opposed to building structures, the designer may choose different loads and supports, even fictitious ones, for different analyses of the same artifact or a portion of it.

In a similar vein, a kinematic analysis may be initiated as soon as the following are defined: the type and size of all the kinematic links; the type of all the joints; and the input motion of the actuator or driving link. At this design stage, the design information is rich enough to perform a kinematic analysis, but not enough to perform a dynamic analysis of the mechanism. An opportunistic dynamic analysis may be triggered at a later design stage as soon as the spatial details of the links and joints, the materials, and the friction conditions between the moving parts are defined.

Notwithstanding the possible complexities, it is possible to envisage the idealization transformations or relationships as active processes, monitoring the development of the product description in the master model for the emergence of analysis opportunities, in the manner of "demons" in Artificial Intelligence programs. (The use of "demons" for collaborative design activities is reported in (Sriram, 2002)).

In order to implement such a capability, we can build on the DBMS literature on coarse-grained transactions, particularly the work of Eastman (Eastman & Kutay, 1991). Eastman and Kutay's representation of a transaction consists of its readset, before constraints, writeset and after constraints of every transaction. The readset and the before constraints determine whether the data available for running a transaction are available and whether all the constraints on the data are satisfied. A transaction is said to be infeasible if the readset is not available or if any of the before constraints are not satisfied.

With the readset and before constraints of a domain-specific functional analysis known, it is possible to determine when that analysis can be invoked during the evolution of a design. It is expected that the characterization of functional models mentioned above can yield this information. Then, as each defining entity or feature is added to the master model, the functionality of the aggregate may be tracked. Once it matches the readset and the before constraints, the analysis can either be initiated automatically or recommended to the designer.

It is assumed that we can define a readset and before constraints for the various functional models characterized by the bases described above, although further research is necessary to validate this approach. Another assumption is that analysis "demons" running in the background can detect readiness for analysis in the master model, not in the idealized functional model for its specific view. This means either that the design information in the master model includes sufficient functional description as the design evolves or, preferably, that the idealization transformations or relationships be implemented as active processes, incrementally building their respective functional models while testing that the readset is complete and the before constraints are satisfied.

## 5 An illustrative example of idealization

This section provides a very simple illustration of some of the idealization relationships between the spatial and functional models discussed with a concrete example. The part being analyzed is an idler arm, a component of a larger mechanical assembly. The part's spatial model is shown in Figure 7.

### 5.1 The models

Three approaches for creating functional models for Finite Element Analysis of this part are shown and compared.

**Figure 7. Idler arm**

The initial functional model shown in Figure 8 is an idealization of the idler arm. A model of this order of complexity may arise early in the preliminary stage of design, as soon as the relative positions of the parts to be connected by the arm become established and the features (sleeves and shafts) approximately proportioned.

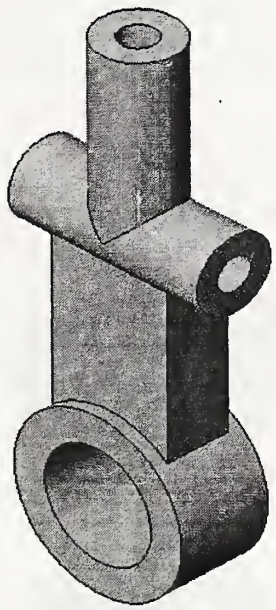

**Figure 8. Initial functional model**

The detailed functional model is a fully detailed representation of the idler arm, as shown in Figure 9. This model was built with functional analysis in mind to insure that idealization would be easy and straightforward. Specifically, the features added to those comprising the initial functional model could be defeatured with one operation (Some analysts may question the inclusion of fillets in the functional model, but the fillets are so constructed in the model that they could be removed by one defeature operation).

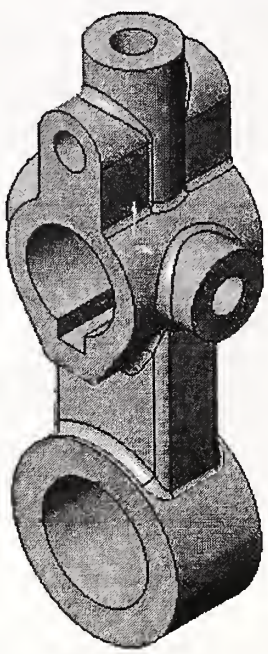

**Figure 9. Detailed functional model**

The third design model considered is also a fully detailed representation of the idler arm; however, in order to demonstrate the additional effort required for manual idealization for analysis if no thought is put into the process of constructing the spatial model, every effort was made to render the idealization of the design model difficult. It can be expected that computer-aided or automatic idealization would prove to be equally difficult. The final design model is the same as the detailed functional model; however, different steps were followed to generate the model.

### 5.2 Model idealization for functional analysis

For finite element analysis, many details of a solid model can be ignored if it is known in advance that they have no effect on the overall results or if only rough estimates of behavior are needed. The initial functional model was built specifically for an approximate analysis and required no defeaturing.

The detailed functional model was built to simulate a two-stage integrated design-analysis scenario. Because of the way the geometry was developed, this model can be idealized in seconds by just suppressing five features. In addition, in order to return to the detailed model, these features only need to be unsupressed.

The third design model was deliberately built poorly to demonstrate the problems that can arise in idealization if no thought is put into developing the spatial CAD model. To extract the idealized model, three features had to be suppressed. The difficult feature is the set of filleted corners of the vertical shaft, which were included as part of the shaft feature before the shaft feature was merged to the two sleeve features. To remove these fillets for an approximate analysis, the underlying shaft feature had to be modified. The total process took about 15 minutes of time for the experienced modeler who had built the original model. The important issue is that an idealized model may contain actual changes in the spatial model, and not just suppressed or unsuppressed features.

### 5.3 Comparison of analysis results

The same analysis was performed on all three models. Loading consisted of a torque in the x-y plane applied to the end of the upper sleeve feature. Material properties of structural steel were used in all analyses.

The meshes for the three models differed slightly. Plots of the finite element meshes for the initial and detailed functional models are shown in Figures 10 and 11.

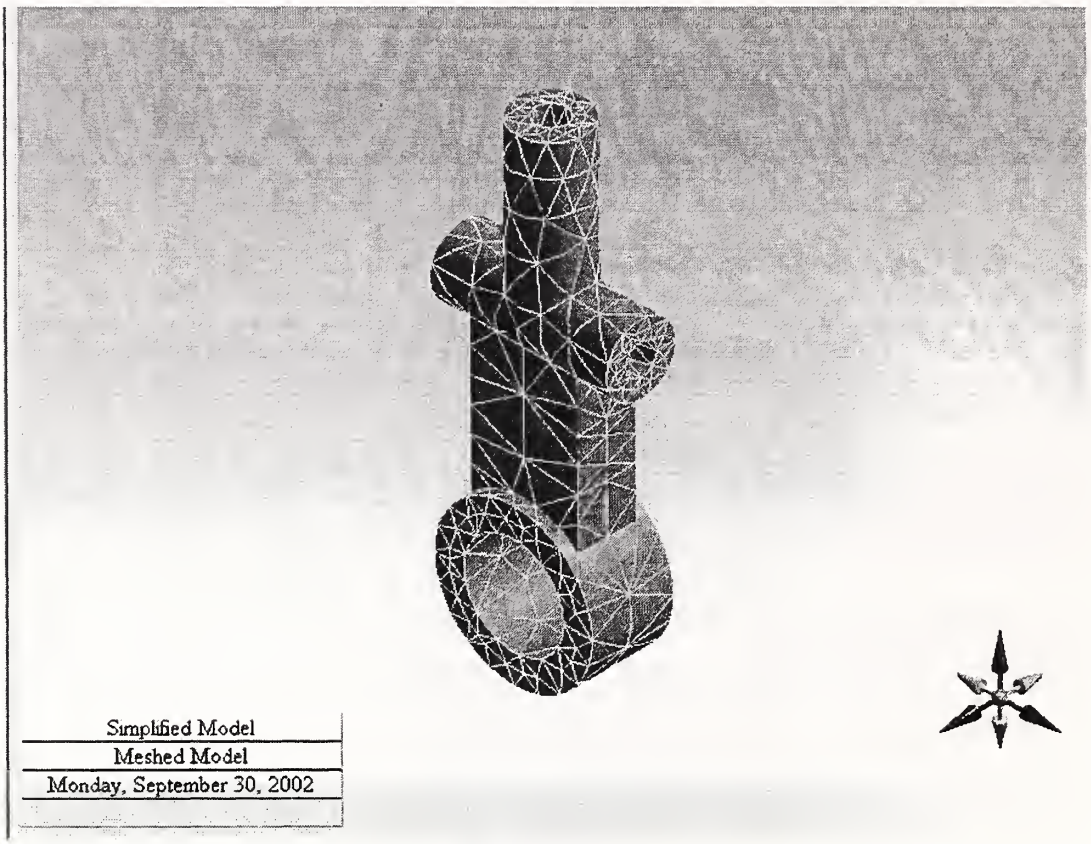


**Figure 10. Mesh of initial functional model**

**Figure 11. Mesh of detailed functional model**

| Parameter | Initial functional model | Detailed functional model | Idealized model from alternate CAD model |
|---|---|---|---|
| Number of features in CAD model | 6 | 11 | 10 |
| Number of nodes | 2786 | 7009 | 2860 |
| Number of elements | 1427 | 3918 | 1485 |
| Run time | 7 seconds | 12 seconds | 6 seconds |
| Max. Von Mises stress | 494 psi | 637 psi | 477 psi |
| Max. displacement | $1.26 \times 10^{-4}$ in | $1.18 \times 10^{-4}$ in | $1.17 \times 10^{-4}$ in |

**Table 1. Comparison of key analysis parameters**

Table 1 presents a comparison of the key parameters of the analyses. As expected, the idealized models accurately predict the detailed model's stiffness, as measured by the maximum displacement. However, they incorrectly predict the location and magnitude of the maximum stress. In the idealized models, the point of maximum stress is at the face of the vertical shaft, as expected for a torsional load. In contrast, in the detailed model the point of maximum stress is at the end of the upper sleeve feature, at the point where the load was applied. This may indicate an idealization error: the inclusion of the full set of features around the upper sleeve makes it considerably stiffer than its counterpart in the idealized model, and thus subject to increased stress. If this were to turn out to be an important design consideration, an alternate idealized model may be used to further investigate the problem.

In summary, in this example idealization of the alternate CAD model required approximately 15 minutes of the analyst's time as opposed to five keystrokes to suppress unneeded features in the spatial model built to anticipate idealization. While this is not a large amount of time, such time can quickly accumulate to hours and even days if many design changes are made and many idealizations are required.

The solution of the detailed functional model required twice the run time of the idealized models. In this example, the run time differences are insignificant because they are a matter of seconds. However, for more complex parts or assemblies, the differences in solution times could become an issue.

A more comprehensive example would involve: multiple functional domains; multiple cycles of design, idealization, functional analysis and mapping; treatment of non-unique mappings; and treatment of spatial conflicts and other incosistencies reulting from multiple mappings.

## 6 Summary and conclusions

The report starts out with a broad analysis and critique of barriersto intimate spatial and functional design integration, and proceeds to recommend one technical step towards the abolition of these barriers: a data architecture that provides *both* a global master model of all aspects of an artifact *and* a set of specialized models of the functional aspects of interest for each of the domains that has a stake in the design and performance of the artifact. Idealization and mapping are defined as the two opposite (but not necessarily inverse) operations, transformations or relationships (depending on one's viewpoint) relating the master model to the domain-specific functional model.

The data architecture presented is at the present a weakly formulated conjecture. The following is a list of activities that may be undertaken to validate the conjecture and, if valid, bring it closer to realization:

- A critical survey of functional engineering domains dealing with spatial and functional design and analysis, as well as of corporate processes related to engineering design and analysis, to: (a) ascertain and catalog the form and function representations used; (b) to identify the information to be shared with other domains and processes; and (c) possibly to develop a formal ontology of the shared information.

- A parallel critical survey of idealization and mapping operations, to: (a) develop a formal classification and taxonomy (and possibly an ontology) of these; (b) identify and suggest replacement of ad-hoc operations that cannot be formalized in the various domains; and (c) identify non-reversible idealization-mapping operations.

- A detailed comparison and evaluation of the three organizing principles presented (and possibly others that may be encountered): net shape; features; and the Core Product Model, i. e., a specific representation of form, function and behavior of an artifact.

- A pilot comparison and evaluation of possible implementation principles, with particular attention to the feasibility of a virtual master model, with the functional models interconnected via executable relationships.

## 7. Acknowledgements and disclaimer

Partial funding for the research reported herein was provided by the TIDE project administered by the Software Engineering Institute at Carnegie Mellon University, Pittsburgh, Pennsylvania.

The material for the illustrative example was prepared by Ms. Anne M Giannetti-Osborn, Technical Services Manager at Mallett Technology, Inc., under a contract to NIST.

The authors benefited from discussions with our colleagues Rachuri Sudarsan and Jay Jung Kim.

The use of any commercial software product or company name in this report is intended to provide readers with information regarding implementations of the research described and does not imply recommendation or endorsement by the National Institute of Standards and Technology.

## References


Andersson, K. (1999), "A design process model for behavior simulations of complex products," in *Proceedings of DETC 1999, Computers in Engineering Conference*, paper no. DETC1999/ CIE-9008, Las Vegas Nevada, USA. September 12-15.

Andersson K., and U. Sellgren (1998), "MOSAIC – Integrated modeling and simulation of physical behavior of complex systems," *Proceedings of NordDesign '98*, Stockholm, Sweden.

Arabshahi, S., D.C. Barton, and N.K. Shaw (1991), "Towards Integrated Design and Analysis," *Finite Elements in Analysis and Design*, Volume 9, Number 4, pp.271-293.

Arabshahi, S., D.C. Barton, and N.K. Shaw (1993), "Steps Towards CAD-FEA Integration," *Engineering with Computers*, Volume 9, Number 1, pp.17-26.

Armstrong, C.G. (1994), "Modeling Requirements for Finite-Element Analysis," *Computer-Aided Design*, Volume 26, Number 7, pp. 573-578.

Armstrong, C.G., R.J. Donaghy, and S.J. Bridgett (1996), "Derivation of appropriate Idealizations in Finite Element Modeling," *Proceedings of the Third International Conference on Computational Structures Technology*, Budapest.

Belaziz, M., A. Bouras, and J.M. Brun (2000), "Morphological Analysis for Product Design," *Computer Aided Design*, Volume 5-6, Number 32, pp. 377-388.

De Martino, T., B. Falcidieno, and S. Habinger (1998), "Design and Engineering Process Integration Through a Multiple View Intermediate Modeler in a Distributed Object-oriented System Environment," *Computer Aided Design*, Volume 30, Number 6, pp. 437-452.

Diaz-Calderon, A., C.J.J. Paredis, and P.K. Khosla (2000), "Organization and Selection of Reconfigurable Models," in Proceedings of the Winter Simulation Conference 2000, Orlando, Florida, December 10-13.

Eastman, C.M. and A. Kutay (1991), "Transaction Management in Design Databases, In Computer-Aided Cooperative Product Development," Sriram, D., Logcher, R., and Fukuda, S. (editors), *Springer Verlag Lecture Notes in Computer Science*, Number 492, Pages 334-351, Springer Verlag.

Fenves, S.J. (2001), "A Core Product Model for Representing Design Information," *NISTIR 6736*. National Institute of Standards and Technology (NIST), Gaithersburg, MD, 38 pp.

Gordon, S. (2001), "An Analyst's View: Step-Enabled CAD-CAE Integration," presented at *NASA's STEP for Aerospace Workshop,* JPL, Pasadena, CA, January 17.

Hoffman, C. M. and R. Joan-Arinyo (1998), "CAD and the Product Master Model," *Computer Aided Design,* Volume 30, Number 11, pp. 905-918.

Hunten, K.A. (1997),"CAD/FEA Integration with STEP AP209 Technology and Implementation," Lockheed Martin Corporation.

Hunten, K.A. (2001), "AP209 ARM Overview," Available at http://pdesinc.aticorp.org/graphics/aps/ap209_tutorial.ppt. [Date visited 7-12-2002].

ISO 10303-1:1994 (ISO,1994a) *Industrial automation systems and integration -- Product data representation and exchange -- Part 1: Overview and fundamental principles.*

ISO 10303-203:1994 (1994b), *Industrial automation systems and integration -- Product data representation and exchange -- Part 203: Application protocol: Configuration controlled design.*

ISO 10303-209:2001 (ISO, 2001), *Product data representation and exchange: Application protocol: Composite and metallic structural analysis and related design.*

Leal, D. (1999), *The Engineering Analysis Core Model: A 'plain man's guide'*, Version 5.0, PDES, Inc.

Mocko, G.M., and S.J. Fenves (2003), "A Survey of Design – Analysis Integration Issues," *NISTIR,* National Institute of Standards and Technology (NIST), Gaithersburg, MD.

OMG (2003), http://www.omg.org/

Pahl, G., and W. Beitz (1996), *Engineering Design: A Systematic Approach*, 2nd ed., Springer-Verlag, London, UK.

Paredis, C.J.J. (2001), "Composable Simulation and Design," Available at http://www-2.cs.cmu.edu/~compsim/.

Peak, R. S., and R.E. Fulton (1994), "A Multi-Representation Approach to CAD/CAE Integration: Research Overview," *Rapid Thermomechanical Design of Electronic Products in a Flexible Integrated Enterprise, Interim Report,* Fulton, R. E.; Ume, C.; et al., Advanced Electronic Packaging Lab., Prj. MS-93-03, Manufacturing Research Center, Georgia Tech, Atlanta, 22-27.

Peak, R. S., and R.E. Fulton (1993a), "Automating Routine Analysis in Electronic Packaging Using Product Model-Based Analytical Models (PBAMs), Part I: PBAM Overview," presented at *ASME Winter Annual Meeting*, paper no. 93-WA/EEP-23, New Orleans, LA, November 28 – December 3.

Peak, R. S., and R.E. Fulton (1993b), "Automating Routine Analysis in Electronic Packaging Using Product Model-Based Analytical Models (PBAMs), Part II: Solder Joint Fatigue Case Studies," presented at *ASME Winter Annual Meeting,* paper no. 93-WA/EEP-24, New Orleans, LA, November 28 – December 3.

Peak, R.S.; R.E. Fulton, I. Nishigaki; and N. Okamoto (1998), "Integrating Engineering Design and Analysis Using a Multi-Representation Approach," *Engineering with Computers*, Volume 14, Number 2., pp. 93-114.

Peak, R.S.; A.J. Scholand, and R.E. Fulton (1996), "On the Routinization of Analysis for Physical Design," 1996 ASME International Mechanical Engineering. Congress & Expo, Application of CAE/CAD to Electronic Systems, EEP-Vol.18, D. Agonafer, et al., Atlanta, pp. 73-82.

Peak, R.S.; Scholand, A.J.; Tamburini, D.R.; Fulton, R.E. (1999) "Towards the Routinization of Engineering Analysis to Support Product Design," Invited Paper for Special Issue: Advanced Product Data Management Supporting Product Life-Cycle Activities, *International Journal of Computer Applications in Technology*, Vol. 12, No. 1, 1-15.

Schlenoff, C., M. Gruninger, F. Tissot, J. Valois, J. Lubell, and J. Lee (2000), "The Process Specification Language (PSL): Overview and Version 1.0 Specification," *NISTIR 6459*. National Institute of Standards and Technology, Gaithersburg, MD.

Sellgren, U. and R. Drogou (1998), "A systems and process approach to behavior modeling in mechanical engineering", in *Proceedings of the Integrated design and Manufacturing in Mechanical Engineering*, IDMME´98, May 27-29, 1998, Campiégne, France.

Sellgren, U. (2000),"An approach to behavior modeling of complex systems in mechanical engineering," *The International ANSYS Conference*, Pittsburgh, PA, USA, August 2000.

Sellgren, U. (2001),"Aspects of simulations in product development," *Endrea-17, Simulation in product development, The Swedish Engineering Design Research and Education Agenda*, Editor. S. Andersson, pp. 1-20.

Shah, J. J. and M. Mäntylä (1995), Parametric and feature-based CAD/CAM : concepts, techniques, and applications, Wiley, N. Y.

Sinha, R., C.J.J. Paredis, and P.K. Khosla (2000), "Integration of Mechanical CAD and Behavioral Modeling," in *Proceedings of the International Workshop on Behavioral Modeling and Simulation*, Orlando, Florida, November 18-21.

Sinha, R., C.J.J. Paredis, and P.K. Khosla (2001), "Interaction Modeling in Systems Design," in *Proceedings of DETC 2001, Computers in Engineering Conference*, paper no. DETC2001/CIE-21285, Pittsburgh, PA, September 9-12.

Shapiro, V., http://sal-cnc.me.wisc.edu/, 2002.

Sriram, R. D., Distributed and Integrated Collaborative Engineering Design, Sarven Publishers, Maryland, 2002.

Szykman, S., S.J. Fenves, S. B. Shooter, and W. Keirouz (2001), "A Foundation for Interoperability in Next-generation Product Development Systems," *Computer-Aided Design*, Vol. 33, No. 7, pp. 545-559.

TIDE (2003), http://www.sei.cmu.edu/tide/

Yoshioka, M. and T. Tomiyama (1997), "Pluggable Metamodel Mechanism: A framework of an Integrated Design Object Modeling Environment," *Computer Aided Conceptual Design `97, Proceedings of the 1997 Lancaster International Workshop on Engineering Design {CACD}'97*, pp. 57-70.